

## **BAB II**

### **KAJIAN TEORI**

#### **A. Deskripsi Skripsi**

Beberapa teori yang digunakan sebagai dasar pada penelitian ini adalah (1) rekayasa perangkat lunak, (2) model perancangan aplikasi, (3) Subag Umum Kepegawaian dan Perlengkapan (UKP), (4) basis data, (5) PHP MySQL, (6) *framework*, (7) *Software Requirements Specification* (SRS), (8) Kerangka Berfikir. Penjelasan mengenai teori-teori tersebut adalah sebagai berikut:

##### **1. Rekayasa Perangkat Lunak**

Menurut Steward (1987), rekayasa perangkat lunak adalah pengelolaan suatu permintaan, teknologi komputer, kemampuan seseorang, waktu, dan biaya untuk membuat suatu produk perangkat lunak sesuai permintaan pelanggan dengan timbal balik sebuah kepuasan kepada pembuatnya. Pressman (2001) mendefinisikan rekayasa perangkat lunak merupakan pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak secara ekonomis yang dipercaya dan dapat bekerja secara efisien pada mesin nyata.

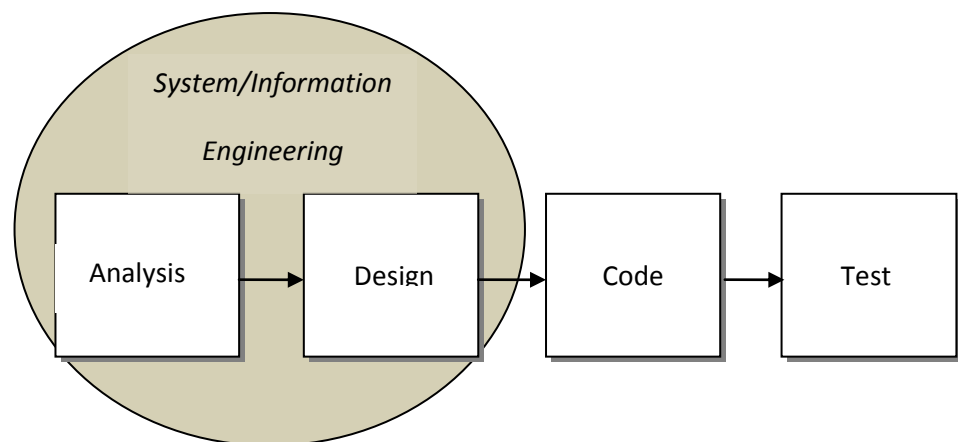
Berdasarkan pendapat kedua ahli di atas, dapat disimpulkan bahwa rekayasa perangkat lunak adalah suatu aplikasi dari sebuah pendekatan yang sistematis, teratur, dan dapat diukur terhadap pengembangan, kinerja, dan pemeliharaan perangkat lunak.

## 2. Model Perancangan Aplikasi

Perancangan aplikasi merupakan gambaran umum untuk menyusun aplikasi yang akan dibuat sebagai pengganti metode yang lama secara keseluruhan dan untuk memperbaiki metode yang telah dibuat. Desain aplikasi yang akan dibuat untuk menggambarkan komponen-komponen aplikasi secara detail. Model perancangan aplikasi yang akan digunakan adalah metode analisis sistem aplikasi terstruktur yakni *Waterfall Model* dan perancangan *Data Flow Diagram* (DFD).

### a. *Waterfall Model*

*Waterfall Model (Classic Life Cycle)* adalah model yang pertama kali muncul yaitu sekitar tahun 1970. Model *Waterfall* sering dianggap kuno, tetapi paling sering digunakan dalam *Software Engineering* (SE).



**Gambar 1. Model *Waterfall* Pressman**

Gambar 1. merupakan tahapan secara umum dari model *Waterfall*.

Pressman (2001: 28) menyatakan bahwa model *Waterfall* diuraikan dengan tahap-tahap sebagai berikut:

- 1) *Analysis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan perancangan aplikasi.
- 2) *Design* adalah tahap penerjemah atau tahap perancangan dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.
- 3) *Code* adalah tahap implementasi dari hasil aplikasi yang telah dirancang dalam bahasa pemrograman yang telah ditentukan dan digunakan dalam pembuatan aplikasi.
- 4) *Test* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian dilakukan agar fungsi-fungsi dalam aplikasi bebas dari *error*, dan hasilnya harus sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

b. Perancangan *Data Flow Diagram* (DFD)

DFD merupakan alat bantu yang digunakan pada metodologi pengembangan sistem terstruktur. Merupakan diagram yang menggunakan simbol-simbol untuk menggambarkan arus data dari sistem yang dikenal dengan nama arus data (*data flow diagram*). DFD dibagi menjadi 2, yaitu:

1) DFD Context

DFD context menggambarkan arus aplikasi pertama kali secara garis besar (top level). Sebelum tahap penguraian dilanjutkan perlu mengidentifikasi urutan/penguraian proses yang ada di aplikasi.

## 2) DFD Leveled

Setelah context diagram dirancang, kemudian akan digambarkan arus data yang lebih terinci lagi yang dinamakan *over view* diagram (level 0) dan seterusnya (Jumi, 2004).

### 3. Subag Umum Kepegawaian dan Perlengkapan (UKP)

Subag Umum Kepegawaian dan Perlengkapan (UKP) adalah sub bagian di fakultas yang mempunyai beberapa tugas penting di fakultas, yaitu diantaranya sebagai berikut:

- a. Menyusun rencana dan program kerja Sub Bagian Umum Kepegawaian dan Perlengkapan.
- b. Menghimpun dan mengkaji peraturan perundang-undangan di bidang ketatausahaan, kerumahtanggaan, kepegawaian, dan perlengkapan.
- c. Mengumpulkan, mengolah, dan menganalisis data ketatausahaan, kerumahtanggaan, kepegawaian, dan perlengkapan.

Tugas UKP di bidang kepegawaian diantaranya adalah sebagai berikut:

- 1) Melakukan urusan persuratan di lingkungan FMIPA.
- 2) Melaksanakan urusan pengelolaan administrasi kepegawaian di lingkungan badan, mempersiapkan rencana kebutuhan pegawai dan usulan mutasi pegawai, pemberhentian serta pensiun pegawai di lingkungan badan.
- 3) Mempersiapkan usulan kenaikan gaji berkala dan kenaikan pangkat pegawai di lingkungan badan.
- 4) Membuat daftar urut kepangkatan, cuti, dan absen pegawai

- 5) Melaksanakan pengurusan kartu kepegawaian antara lain KARPEG, KARIS, KARSU, dan TASPEN.
- d. Melakukan urusan kearsipan di lingkungan FMIPA.
  - e. Melakukan pemeliharaan keamanan dan ketertiban lingkungan.
  - f. Melakukan urusan penerimaan tamu pimpinan.
  - g. Mempersiapkan sarana pelaksanaan rapat dinas, upacara resmi dan pertemuan ilmiah di lingkungan FMIPA.
  - h. Melakukan urusan inventarisasi dan pengelolaan barang milik Negara.
  - i. Melakukan penyusunan instrument pemantauan kegiatan ketatausahaan, kerumahtanggaan, kepegawaian, dan perlengkapan.
  - j. Melakukan penyimpanan dokumen dan surat di bidang ketatausahaan, kerumahtanggaan, kepegawaian, dan perlengkapan.
  - k. Melaksanakan tugas lain yang diberikan oleh atasan.

Peminjaman ruang diurus oleh UKP bertujuan untuk memberikan fasilitas mahasiswa dan dosen untuk dapat melangsungkan kegiatan-kegiatan di luar kegiatan perkuliahan. Peminjaman ruang dicatat oleh petugas bagian UKP untuk menentukan jadwal dan ruangan yang akan digunakan oleh masing-masing kegiatan.

#### 4. **Basis Data (*Database*)**

Fathansyah (2004) menjelaskan bahwa basis data dapat didefinisikan dalam beberapa sudut pandang, yakni: himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Kumpulan data yang saling

berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis. Menurut Ramez Elmasri (Dzacko: 2007) basis data dibatasi pada arti implisit secara khusus, yaitu:

- 1) Basis data merupakan penyajian suatu aspek dari dunia nyata.
- 2) Basis data merupakan kumpulan data dari bermacam-macam sumber secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tidak mempunyai arti, tidak dapat disebut basis data.
- 3) Basis data perlu dirancang, dibangun dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa *user* (pengguna) dan beberapa aplikasi yang sesuai dengan kepentingan *user* (pengguna).

a. Model Basis Data Relasional

Model basis data relasional merupakan model yang paling sederhana sehingga mudah digunakan dan dipahami oleh pengguna. Model ini menggunakan sekumpulan tabel dengan masing-masing tabel tersusun atas sejumlah baris dan kolom. Pada model data relasional, keterkaitan antara dua buah tabel disebut relasi. Hal-hal yang perlu diperhatikan dalam penyusunan tabel-tabel dalam basis data relasional adalah atribut kunci. Penentuan kunci merupakan hal yang penting dalam basis data relasional kunci. Kunci sebagai metode untuk mengakses suatu baris tertentu sekaligus menjadi pengenalan unik (mempunyai ciri khas) rekaman di dalam

suatu tabel. Kunci dapat berupa sebuah atribut atau gabungan dari beberapa atribut.

Menurut Andri K. (2003: 76-77), atribut kunci dibagi menjadi:

1) Kunci Kandidat (*Candidat Key*)

Kunci kandidat adalah suatu atribut yang mengidentifikasi secara unik dari suatu kejadian yang sifatnya khusus dari suatu entiti. Kunci kandidat ini tidak ganda.

2) Kunci Primer (*Primary Key*)

Kunci primer adalah kunci kandidat yang dipilih untuk mewakili setiap kejadian dari suatu entiti.

3) Kunci Alternatif (*Alternative Key*)

Kunci alternatif adalah kunci kandidat yang tidak dipilih sebagai kunci primer. Kunci alternatif lebih orientasi untuk pengurutan rekaman di dalam sebuah tabel atau entiti.

4) Kunci Tamu (*Foreign Key*)

Kunci tamu adalah kunci primer suatu entiti yang dihubungkan pada entiti lain. Biasanya jika ada dua file, kunci primer akan diletakkan pada entiti induk, sedangkan pada entiti anak diletakkan kunci tamu.

b. Database Management System (DBMS)

Budi Raharjo (2011) mendefinisikan *Database Management System* (DBMS) adalah kumpulan program yang digunakan untuk mendefinisikan, mengatur dan memproses database. Sedangkan database adalah sebuah struktur yang dibangun untuk keperluan penyimpanan data. DBMS

merupakan alat yang berperan untuk membangun struktur tersebut. Pengelolaan basis data secara spesifik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (software) yang khusus/spesifik. Perangkat inilah yang disebut DBMS yang menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. DBMS menerapkan mekanisme pengamanan data secara bersama, pemaksaan keakuratan/konsistensi data dan sebagainya (Syahril Akbar, 2010: 46). Lukmanul Hakim (2007) menyatakan bahwa tujuan dari DBMS adalah mempermudah dalam membuat struktur data dan membebaskan pemrogram dari masalah penyusunan file. DBMS telah berkembang dari suatu konsep database sebagai sesuatu yang jelas berbeda dengan program yang mengakses. Informasi fisik disimpan dan diambil dari database oleh DBMS. Sedangkan DBMS yang modern secara langsung dapat mencari record di dalam tabel yang memiliki banyak record. Sementara DBMS mengolah semua informasi di dalam database, aplikasi bisa mengakses informasi ini melalui pernyataan SQL (*Structured Query Language*), sebuah bahasa yang mampu mengakses database dengan operasi tingkat tinggi.

## **5. PHP MySQL**

Lukmanul Hakim (2007) menyatakan bahwa PHP merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server (*server-side scripting*). Kemudian hasilnya dikirimkan ke klien/browser dalam bentuk HTML.



Sidik (2003) menyimpulkan bahwa prinsipnya, php dapat dikerjakan oleh program CGI di server, seperti mendapat data dari form, menghasilkan isi halaman web yang dinamis, menerima cookies dan session. Namun, kemampuan PHP yang paling diandalkan dan signifikan adalah dukungannya terhadap banyak database, seperti MySQL, Postgres, Oracle, Access, Informix, DB2, Dbase, dan lain-lain. Sehingga PHP dapat membuat halaman web yang menggunakan data dari database dengan sangat mudah.

Sidik (2003) menyatakan bahwa semua perintah yang ada pada SQL dapat dilakukan pada database MySQL, seperti input atau menambah data, menjalankan query, mengambil data dari database dan menampilkannya, mengubah data, dan menghapus data dari database. MySQL sebagai database server dapat dikatakan lebih unggul dibandingkan database server lainnya dalam melakukan query terhadap data, sehingga MySQL sangat cocok digunakan untuk keperluan di halaman web yang membutuhkan akses data yang cepat.

MySQL merupakan database server yang memiliki konsep database modern, selain itu MySQL memiliki banyak keistimewaan, antara lain:

- a. *Portability*, artinya dapat berjalan stabil pada berbagai sistem operasi.
- b. *Open Source*, artinya bersifat free/gratis dan dapat dikembangkan source codenya secara bebas.
- c. *Performance running*, artinya memiliki kecepatan kinerja yang handal.
- d. *Scalability dan Limite*, artinya mampu menangani database dalam skala besar.

- e. *Connectivity*, artinya mudah dikoneksikan dengan berbagai pemrograman.
- f. *Security*, artinya sebagai database server yang diakses dalam jaringan Internet, maka tingkat keamanannya pun harus tinggi (Lukmanul Hakim, 2007).

## 6. Framework

*Framework* adalah desain yang bisa digunakan kembali dan biasanya dinyatakan sebagai satu sel abstraksi *class* yang mengatur hubungan antar *class*. Perancangan pada *framework* dibuat sedemikian rupa sehingga sebagian atau seluruh *software* dapat digunakan kembali. *Framework* merupakan kerangka kode yang dapat disimpulkan dengan *classes* yang spesifik atau dengan fungsi yang telah dirancang untuk mengatasi masalah yang dihadapi (Johan I, dkk, 2011: 7).

*Codeigniter* adalah *framework* untuk *web* yang dibuat dalam format PHP. *Codeigniter* bersifat *opensource* yang gratis untuk digunakan dan dapat dikembangkan secara ilegal. *Codeigniter* berlisensi *Apache/ BSD-style open source* yang memperbolehkan penggunaan dari *framework* ini sesuai keinginan (Jefri, 2013:3).

Jefri (2013) menyatakan bahwa MVC adalah sebuah *software* yang memisahkan antara aplikasi logika dengan presentasi pada halaman *web*. Sehingga hal ini akan menyebabkan halaman web akan mengandung sedikit kode program karena sudah terjadi pemisahan antara tampilan dan pemrograman.

- a. **Model**, merupakan struktur data. Secara spesifik class model akan mengandung fungsi kode yang akan membantu dalam segala proses yang berhubungan dengan *database* seperti memasukkan, mengedit, mendapatkan, dan menghapus dalam sebuah *database*.
- b. **View**, merupakan informasi yang disampaikan ke pengguna. Sebuah *view* biasanya berupa halaman *web*, tetapi dalam sebuah *view* juga bisa berupa sebuah fragmen halaman seperti *header* dan *footer*. View juga bisa berupa halaman *RSS* atau jenis halaman *web* yang lain.
- c. **Controller**, merupakan sebuah perantara antara *Model* dan *View* serta semua sumber yang dibutuhkan untuk memproses permintaan *HTTP* dan dalam membuat halaman *web*.

## 7. *Software Requirements Specification (SRS)*

Menurut Wisnu (2012), menyatakan bahwa *Software Requirements Specification (SRS)* adalah dokumen yang menjelaskan tentang berbagai kebutuhan yang harus dipenuhi oleh suatu *software*. Dokumen ini dibuat oleh *developer* (pengembang *software*) setelah menggali informasi dari calon pemakai *software*. Pembuatannya mengikuti standar yang ada dan paling dianggap benar oleh para praktisi rekayasa *software* di dunia. Oleh karena itu, standar yang akan dibahas di sini adalah standar dari IEEE, singkatan dari *Institute of Electrical and Electronics Engineers*.

SRS harus bermanfaat bagi pengguna, penyedia, atau perorangan. Adapun manfaat dari SRS antara lain sebagai berikut:

- a. Sebagai bentuk perjanjian antara pengguna dan penyedia tentang software apa yang akan dibuat.
- b. Mengurangi beban dalam proses pengembangan software.
- c. Sebagai bahan perkiraan biaya dan rencana penjadwalan.
- d. Sebagai dasar validasi dan verifikasi software di ujung penyelesaian proyek nantinya.
- e. Memfasilitasi transfer, semisal software tersebut ingin ditransfer ke pengguna atau mesin-mesin yang lain. Menggunakan lebih mudah jika ingin mentransfer software ke bagian-bagian lain dalam organisasinya. Bahkan, jika terjadi pergantian personil pengembang, proyek dapat mudah ditransfer ke personil baru dengan memahami SRS ini.
- f. Mendasari perbaikan produk software dikemudian hari. Jadi, SRS boleh diperbaiki dengan alasan dan mekanisme tertentu serta atas kesepakatan antara pengunadan pengembang.
- g. Dengan menggunakan SRS, pengguna dapat menuangkan semua ide terkait software dengan jelas dan akurat sehingga pengembang dapat memahami apa yang diinginkan pengguna dengan tepat. Standar ini akan membantu dalam mengembangkan *outline* SRS yang baku untuk pengguna, membantu membuat dokumen SRS dengan format dan isi yang standar (minimal), serta membantu mengembangkan rincian-rincian pendukung lainnya.

SRS harus memiliki jaminan kualitas perangkat lunak dan pengujian aplikasi. Jaminan perangkat lunak adalah aktivitas pelindung yang

diaplikasikan pada seluruh proses perangkat lunak. Tujuan dari jaminan kualitas adalah untuk memberikan data yang diperlukan oleh manajemen dan menginformasikan masalah kualitas produk, sehingga dapat memberikan kepastian dan konfidensi bahwa kualitas produk dapat memenuhi sasaran, tidak hanya berkualitas menurut pengembang tapi juga berkualitas dan sesuai dengan keinginan pengguna (Nastiti, 2012: 35).

Menurut McCall dalam Nastiti (2012: 36), faktor-faktor penentu kualitas perangkat lunak adalah sebagai berikut:

- a. *Correctness*, sejauh mana suatu perangkat lunak memenuhi spesifikasi dan tujuan penggunaan perangkat lunak dari user.
- b. *Reliability*, sejauh mana keakuratan suatu perangkat lunak dalam melaksanakan fungsinya.
- c. *Efficiency*, banyaknya kode program yang dibutuhkan suatu perangkat lunak untuk melakukan fungsinya.
- d. *Integrity*, sejauh mana akses ke perangkat lunak dan data oleh pihak yang tidak berhak dapat dikendalikan.
- e. *Usability*, usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan input, dan mengartikan output dari perangkat lunak.
- f. *Maintainability*, usaha yang diperlukan untuk menetapkan dan memperbaiki kesalahan dalam program.
- g. *Testability*, usaha yang diperlukan dalam pengujian program untuk memastikan bahwa program melaksanakan fungsi yang ditetapkan.

- h. *Flexibility*, usaha yang diperlukan untuk memodifikasi program operasional.
- i. *Portability*, usaha yang diperlukan untuk memindahkan program dari perangkat keras/lingkungan sistem perangkat lunak tertentu ke yang lainnya.
- j. *Reusability*, tingkat kemampuan program/bagian dari program yang dapat dipakai ulang dalam aplikasi lainnya, berkaitan dengan paket dan lingkup dari fungsi yang dilakukan oleh program.
- k. *Interoperability*, usaha yang diperlukan untuk menggabungkan satu sistem dengan yang lainnya.

Pengujian *software* adalah metode yang dilakukan untuk menjelaskan tentang pengoperasian perangkat lunak yang terdiri dari perangkat pengujian, metode pengujian dan pelaksanaan pengujian. Ada 2 jenis pengujian, yaitu:

**a. Pengujian *Alpha***

Pengujian *Alpha* dilakukan pada sisi pengembang. Software digunakan pada pengaturan yang natural dengan pengembang yang memandang sisi pemakai dan merekam semua kesalahan dan masalah pemakaian. Pengujian *Alpha* bertujuan untuk mengidentifikasi dan menghilangkan sebanyak mungkin masalah sebelum akhirnya sampai ke pengguna, dilakukan setelah *software* selesai oleh orang-orang yang tidak terlibat dalam pengembangan dan memang ahli dibidangnya menggunakan formulir evaluasi resmi.

b. Pengujian *Betha*

Pengujian *Betha* dilakukan pada satu atau lebih pengguna *software* dalam lingkungan yang sebenarnya, pengembang tidak terlibat pengujian ini. Pengguna merekam semua masalah (nyata atau imajiner) yang ditemui selama pengujian dan melaporkan pada pengembang pada interval waktu tertentu. Berdasarkan hasil pengujian *Betha*, dicari presentase masing-masing jawaban dengan menggunakan rumus:

$$Y = (P/Q) \times 100\%$$

Keterangan:

Y = Nilai presentase

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

## 8. Kerangka Berpikir

Proses peminjaman ruang di UKP FMIPA terdapat beberapa kendala yang menyebabkan proses peminjaman ruang terhambat. Oleh karena itu, perlu dibuat sebuah aplikasi yang mampu mendukung proses peminjaman ruang agar proses peminjaman ruang dapat lebih baik. Salah satunya adalah aplikasi peminjaman ruang di UKP FMIPA UNY sehingga semua kegiatan baik perkuliahan maupun kegiatan di luar perkuliahan dapat terorganisir dengan baik. Sebuah aplikasi berbasis *website* yang menyediakan aplikasi peminjaman ruang untuk kegiatan-kegiatan di FMIPA UNY.

Aplikasi ini diharapkan mampu memberikan kemudahan baik bagi para mahasiswa maupun dosen untuk melakukan peminjaman ruang di FMIPA UNY. Dengan aplikasi ini, dokumentasi peminjaman ruang akan lebih tertata dan mudah dilakukan serta memungkinkan peminjam untuk

melakukan peminjaman ruangan yang siap dipakai sesuai dengan kebutuhan hari, tanggal, dan jam peminjaman tanpa harus bertatap muka untuk menanyakan jadwal penggunaan ruang yang siap pakai pada bagian UKP FMIPA UNY.